(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 942 569 A2

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**15.09.1999 Bulletin 1999/37**

(51) Int Cl.[6]: **H04L 29/06**

(21) Application number: **99301555.1**

(22) Date of filing: **02.03.1999**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **13.03.1998 US 39112**

(71) Applicant: **LUCENT TECHNOLOGIES INC.**
**Murray Hill, New Jersey 07974-0636 (US)**

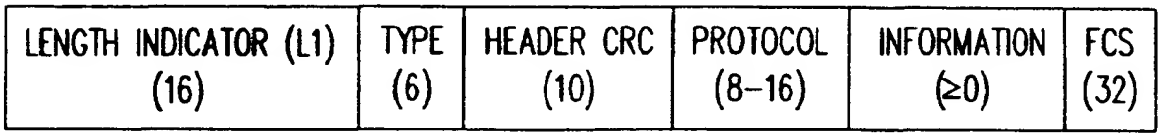
(72) Inventors:
- **Doshi, Bharat Tarachand**
  **Holmdel, New Jersey 07733 (US)**
- **Dravida, Subrahmanyam**
  **Freehold, New Jersey 07728 (US)**
- **Hernandez-Valencia, Enrique**
  **Highlands, New Jersey 07732 (US)**
- **Matragi, Wassim A.**
  **Brooklyn, New York 11209 (US)**
- **Qureshi, Muhammed Akber**
  **Metuchen, New Jersey 08840 (US)**

(74) Representative:
**Watts, Christopher Malcolm Kelway, Dr. et al**
**Lucent Technologies (UK) Ltd,**
**5 Mornington Road**
**Woodford Green Essex, IG8 0TU (GB)**

## (54) Simple data link (SDL) protocol

(57) A simple point-to-point data link protocol (SDL) is defined which is based on the use of a length indicator field and an error check field, rather than a flag, for performing packet boundary recovery in a receiver. In an embodiment of the invention, an SDL transmitter transmits SDL packets comprising a header and a variable length payload. The SDL header comprises a length indicator (LI) field, a type field and a cyclic redundancy check (CRC) field. For receiving these transmitted SDL packets, SDL supports the use of a self-synchronization/self-delineation technique in the receiver. The receiver performs self-delineation as a function of the LI field, and performs self-synchronization, or packet recovery, as a function of both the LI field and the header CRC field. In particular, in performing packet recovery, the receiver performs a CRC check over each received SDL packet header and synchronization is declared after $N$ correct checks, e.g., $N = 4$. The SDL receiver operates in a hunt mode when performing synchronization, and a normal mode when synchronization has been accomplished.

*FIG. 1*




EP 0 942 569 A2

Printed by Jouve, 75001 PARIS (FR)

unscramble the SDL PDU. This form of hunt mode provides for extra protection against potential malicious users.

## Brief Description of the Drawing

**[0008]**

FIG. 1 shows an illustrative SDL frame in accordance with the principles of the invention;
FIG. 2 shows a packet communications system in accordance with the principles of the invention;
FIG. 3 shows an illustrative SDL packet stream;
FIG. 4 shows SDL receiver states in accordance with the principles of the invention;
FIG. 5 shows an illustration of SDL receiver processing in the hunt state;
FIG. 6 shows an illustrative flow chart for recovering packet boundaries in accordance with the principles of the invention; and
FIG. 7 shows another embodiment of the inventive concept.

## Detailed Description

**[0009]** In accordance with the inventive concept, a new point-to-point data link layer protocol, referred to herein as the Simple Data Link (SDL) protocol is described. SDL does not use flags for delineation of protocol data units (PDU) over the data link. Instead, SDL uses a self-synchronization/self-delineation technique where a CRC check is performed over the SDL packet header and synchronization/delineation is declared after few correct checks (described below). It is this avoidance of byte-level processing that makes SDL particularly scaleable to very high link rates. The packet framing and synchronization mechanisms for SDL are described below.

### SDL Framing

**[0010]** SDL framing is designed to support both the multiplexing of multiprotocol datagrams as well as the multiplexing of a small number of logical virtual links within the data link. A summary of an illustrative SDL frame structure is shown in FIG. 1. (This assumes PPP as the framed PDU). Fields are to be transmitted from left to right. An SDL frame comprises a header and a PDU. The header comprises a length indicator, type, and header CRC fields. The PDU comprises a protocol, information, and frame check sequence (FCS) field.

**[0011]** The length indicator (LI) Field is 2 octets (16 bits) long. Its value indicates the total length of the data link PDU, including headers, information and trailing PDU fields.

**[0012]** The type field is 6 bits long. Initially, the use of the type field is "reserved." However, the value of the type field may be used for a variety of functions. For example, identifying the type of service to be associated with the datagram encapsulated in the information field; providing support for simple control functions; providing an indication that the encapsulated datagram contains link control information; identifying logical channels within a virtual SDL link, or identifying multiple physical SDL links multiplexed into a single virtual SDL link, via logical channel identifiers.

**[0013]** The header CRC field is intended for single bit error correction and multiple bit error detection (described below). The header CRC field is 10 bits long and its value indicates the coefficients of the header integrity check. The header CRC field is calculated over all bits in the length indicator and type fields.

**[0014]** The protocol field is either one or two octets long. Its value identifies the protocol type of the datagram encapsulated in the information field. The structure of this field is the same as described for the Protocol Field for PPP (e.g., see W. Simpson, *"PPP in HDLC-like Framing,"* RFC 1662, July 1994).

**[0015]** The information field is zero or more octets long. It contains the datagram from the protocol field identified in the protocol field. The default maximum value for the information field, also referred to as the Maximum Receive Unit (MRU), is 1500 bytes. The MRU may also be negotiated.

**[0016]** The frame check sequence (FCS) Field is 4 octets long and constitutes the trailer of the SDL frame. Its value indicates the coefficients for the frame integrity check. The FCS field is calculated over all bits in the protocol and information Fields. The FCS field provides payload protection against data link errors. For real time services, error checking of the payload may not be necessary. In that event, there are two options. The real time nature is indicated by a setting of the type field. At the receiver, these packets are handed to the next layer even if the FCS fails, in which case, an indication that the FCS has failed is also passed along. A second option is not to include the FCS field for real time services, which are indicated by a setting of the type field.

**[0017]** In accordance with the inventive concept, SDL supports packet length delineation through the LI field, and packet boundary recovery functions through both the LI field and the header CRC field during link synchronization procedures (described below).

### SDL Link Operation

**[0018]** An illustrative packet communications system 100 in accordance with the principles of the invention is shown in FIG. 2. Other than the inventive concept, the elements shown in FIG. 2 are well-known and will not be described in detail. For example, modulator 110 forms a transmission signal at the physical layer for transporting the data as known in the art, e.g., by modulating a carrier using quadrature amplitude modulation (QAM). Also, although the illustrative embodiment is representative of a PPP connection between, e.g., res-

**[0027]** Once SDL deformatter 160 finds a valid CRC, the value of the Length Indicator field is examined and based on that value, the next CRC is examined for this potential frame boundary. Yet, the SDL deformatter 160 continues to slide one byte at a time. A separate counter is initialized for each potential frame boundary detected (described further below). The practical number of such counters to be supported is bounded by the maximum frame size.

**[0028]** This frame re-synchronization algorithm requires that enough correct CRCs are detected consecutively, say $N$ ($N>1$), to provide a low probability of a false boundary delineation. When the first sequence of such $N$ consecutive and valid CRC are encountered then the SDL receiver transitions back to the sync state (as shown in FIG. 4). If any of the header CRC checks fails during this state the SDL receiver adjusts the counters used in the hunting procedure according to the last valid CRC match (described further below).

**[0029]** With the proposed re-synchronization algorithm, 4 consecutive matches of SDL header CRCs (where the second, third and fourth SDL header locations are derived from the length indicators of previous headers) should suffice to guarantee a low frame re-synchronization failure for most practical scenarios of interest. Storing the length indicators in counters while continuing to performing the CRC computation by sliding one byte at a time has the advantage of resolving the true SDL header location versus an accidental match of the CRC in the shortest possible time. With 4 consecutive matches, a reliability of $2^{(-40)}$ or $10^{(-12)}$ can be provided. The proposed frame re-synchronization procedure guarantees that packet boundary re-synchronization is achieved in exactly 4 packet intervals.

**[0030]** If the reliability needed is $10^{(-9)}$ then 3 consecutive passes of the CRC could be used to declare packet boundary acquisition. When there are random errors, the header CRC corrects single bit errors in the SDL header. As noted, the SDL receiver enters the hunt state in presence of burst errors. Generally burst errors in fiber systems appear to last between 20 to 40 ms. An additional re-synchronization time of 4 packet intervals is insignificant at these transmission speeds. Another option is to pass up to the higher layer (not shown) packets that pass two consecutive CRC checks. If, in fact these packets are erroneous then the 32 bit FCS on the payload of the packet will detect and discard the packets.

**[0031]** An illustrative flow chart for reacquiring packet boundaries in accordance with the principles of the invention is shown in FIG. 6. For the purposes of this description it is presumed that SDL deformatter 160 comprises a stored-program-controlled processor and is suitably programmed to carry out the below-described method using conventional programming techniques, which, as such, will not be described herein. The process starts in step 505. If SDL deformatter 160 does not detect an uncorrectable error in the current received SDL packet, SDL deformatter remains in the sync state in step 510 and continues to recover payloads from received SDL packets using the respective values of the LI field.

**[0032]** However, if SDL deformatter 160 detects an undetectable error, SDL deformatter 160 enters the hunt state in step 515 and initializes a set of variables:

$i = 0$; a counter; and
$j = 0$; a counter.

- $L$ is an array of counters that is initialized with the length indicator of valid CRCs. The size of the array is determined by the number of "sequences" of valid headers which has a theoretical upper bound of the maximum packet size allowed (e.g., 5000 bytes). However, with a high degree of confidence($7 \times 10^{-24}$), only 24 sequences need to be pursued.
- $K$ is a set of indices of array $L$ which contains the sequences of valid headers (it is recommended that the maximum number of elements be 24 for the reasons explained above).
- $L[k]$ is a counter indicating the remaining number of bytes to the next "expected header" of sequence $k$.
- $x$ is an element of the set $K$ (a sequence element);
- $R$ is an array of counters keeping track of the number of valid headers in a sequence;
- $R[x]$ is a counter indicating the number of valid headers in the sequence x, $0 \leq R[x] \leq 4$.

**[0033]** A valid sequence means a sequence of headers where the length indicator of the $i^{th}$ header points to the starting location of the $(i+1)^{th}$ header.

**[0034]** In step 520, SDL deformatter 160 evaluates the CRC. If the CRC does not check, then SDL deformatter 160 goes to step 540, described below. If the CRC does check, SDL deformatter 160 increments the values of $i$ and $j$ in step 525. In step 530, SDL deformatter 160 determines the next length in $L[i]$. In step 540, SDL deformatter 160 slides by one byte, and for all $k \in$ K decrements $L[k]$ by one and checks the respective possible CRC field values. In step 545, SDL deformatter 160 checks the CRC for passing. If this CRC check does not pass, SDL deformatter 160 returns to step 540 via step 535 and slides an additional byte. (In step 535, if $L[x] = 0$ for any $x \in$ K, the set K is reduced by {x}.) If the CRC does check, SDL deformatter 160 checks for any $x \in K - \{0\}$ if $L[x] = 0$ in step 550. If no, SDL deformatter 160 increments the value of j and updates the value of $i$ in steps 555 and 560, respectively. Additionally SDL deformatter 160 changes the set $K$ to be equal to $K \cup \{i\}$ in step 560. SDL deformatter 160 then returns to step 530. However, if in step 550 the result is yes, SDL deformatter 160 increments the value of the reliability counter, $R[x]$ in step 565. In step 570, SDL deformatter 160 checks if the value of the reliability count is equal to four. If yes, synchronization is declared in step 575

nal 656 in performing the above-described packet boundary recovery. (It should be noted that alternative implementations could also be used, e.g., providing one input signal to the modulator and controlling. e.g., a multiplexer which selects either the output signal from formatter 610 or scrambler 620 for application to modulator 615. Similar alternative implementations can be performed in the corresponding receiver.)

**[0041]** The foregoing merely illustrates the principles of the invention and it will thus be appreciated that those skilled in the art will be able to devise numerous alternative arrangements which, although not explicitly described herein, embody the principles of the invention and are within its scope. For example, although the inventive concept was illustrated herein as being implemented with discrete functional building blocks, e.g., an SDL formatter. etc., the functions of any one or more of those building blocks can be carried out using one or more appropriately programmed processors, e.g., a digital signal processor; discrete circuit elements; integrated circuits; etc.

**[0042]** For example, it should be noted that if 4 byte or 8 byte parallel processing is deemed necessary then all of the operations shown in FIG. 6 including CRC computations can be performed in parallel. Also, although not described, Link Configuration Protocol (LCP) procedures may be defined for SDL. Such procedures should be consistent with existing configuration capabilities in the LCP for PPP. SDL specific parameters, such as interpretation of the type field, SDL header compression, protocol field compression, etc. may be configured using such LCP procedures. Most existing NCP (network control protocols) should work over SDL with minimal (if any) modifications.

**[0043]** In addition, the SDL packet delineation mechanism can be used even when the physical layer does not provide byte boundaries. An example is the mapping of IP packets using SDL directly onto optical wavelengths. In this case, during the hunt state the CRC checking needs to be done by sliding one bit at a time.

## Claims

1. A method for use in performing packet boundary recovery, the method comprising the steps of:

   receiving a signal representing a stream of packets, each packet comprising a header portion and a payload portion, the header portion further comprising a length indicator field and an error check field; and
   responsive to an error indication in the packet header, performing packet boundary recovery as a function of data representative of a length indicator field value and data representative of an error check field value.

2. The method of claim 1 wherein the performing step includes the step of declaring packet synchronization after N correct checks of values of the error check field

3. The method of claim 1 wherein the error check field is a cyclic redundancy check field.

4. The method of claim 1 further comprising the step of detecting the error indication in the packet header.

5. The method of claim 1 wherein the performing step includes the step of disabling a descrambler.

6. Apparatus for use in packet equipment, the apparatus comprising:

   a demodulator for receiving a signal representing a stream of packet data, each packet comprising a header portion and a payload portion, the header portion further comprising a length indicator field and an error check field;
   a descrambler, responsive to an output signal from the demodulator for descrambling at least the payload portion of each received packet;
   a deformatter having a first state of operation and a second state of operation, wherein in the first state the deformatter is responsive to a value of each length indicator field as represented in the output signal of the demodulator for delineating packet boundaries for providing the descrambled payload portion of each packet, and wherein in the second state of operation the deformatter performs packet boundary recovery by scanning the packet data as represented in the output signal of the demodulator for N valid packet headers as represented by associated length indicator and error check field values.

7. The apparatus of claim 6 wherein the error check field is a cyclic redundancy check field.

8. Apparatus for use in a packet system in which a receiver performs packet boundary recovery as a function of a value of a length indicator field and an error check field the apparatus comprising:

   a formatter for forming packets, each packet comprising a header portion and a payload portion, the header portion further comprising the length indicator field and the error check field;
   a scrambler for scrambling the payload portion of each packet; and
   a modulator for modulating (a) the packet headers, which are not scrambled, and (b) the scrambled packet payloads.

*FIG. 4*

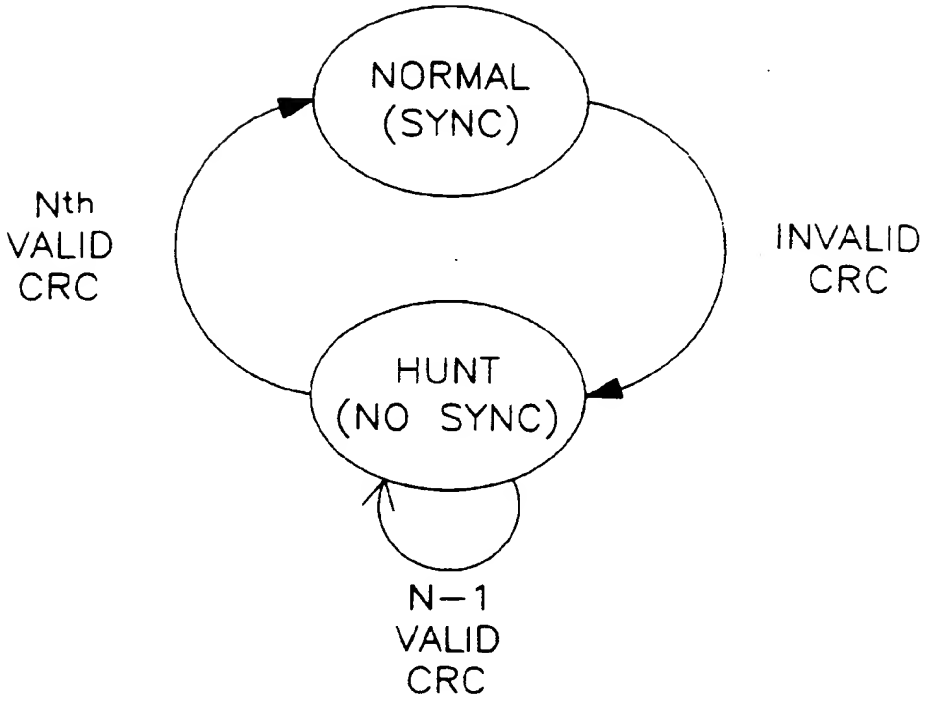



*FIG. 5*

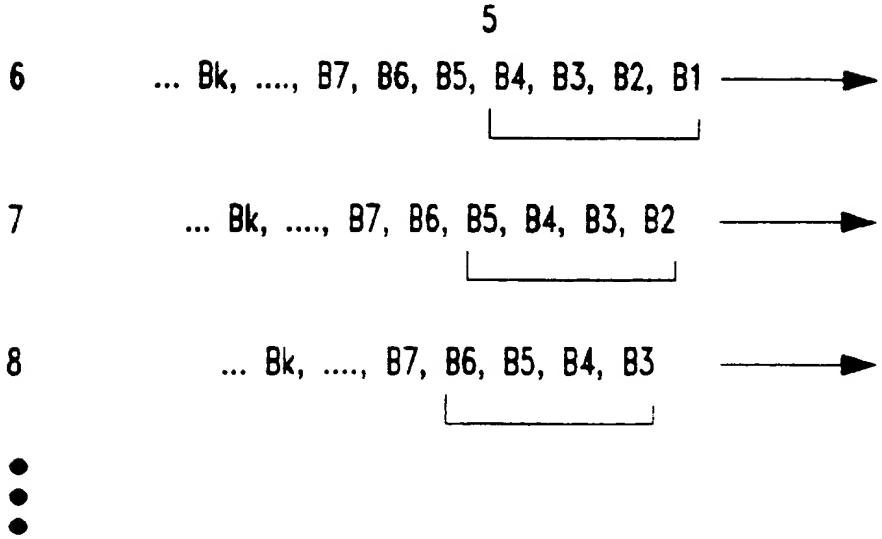


FIG. 7